Une interface gestuelle pour l'apprentissage de la rythmique

Kamp Jean-François[1] Ménier Gildas[1] Sylvie Gibet[1]

[1]Université Européenne de Bretagne – laboratoire IRISA-UBS

Centre de Recherche Yves Coppens
Campus Tohannic
56000 Vannes
(jean-francois.kamp, gildas.menier, sylvie.gibet)@univ-ubs.fr

Résumé

Le système pédagogique d'apprentissage de la rythmique que nous présentons dans ce papier relève à la fois de l'interaction gestuelle et de l'instrument de musique virtuel. Il utilise le gant de données CyberGlove® comme modalité en entrée pour piloter une application musicale d'enregistrement, de production et de modification de sons de percussion. Le papier présente les principes de base du système qui consiste à générer des sons de percussion par reconnaissance de gestes de la main. Cette reconnaissance de la gestuelle est obtenue en temps réel par une méthode de type logique floue.

Mots Clef

Apprentissage du rythme, pédagogie musicale, interaction gestuelle, instrument virtuel, reconnaissance de gestes, logique floue, interface musicale, Gant de données CyberGlove®.

Abstract

In this paper, we present a gestural interface built to support music pedagogy : the training of the rhythm for drummers. The data gloves CyberGlove® is used in an augmented reality system both to sample rhythm in a natural way and to control the production and recording of drum patterns. This paper gives some guiding principles about the overall system where the main idea consists in producing drum sounds through hand gesture recognition. The hand gestures are recorded in real time using a glove device and are recognized by fuzzy logic method.

Keywords

Rhythm learning, music pedagogy, gesture interaction, music interface, Virtual Musical Instrument, gesture recognition, CyberGlove® input device, fuzzy logic.

1 Contexte

Le cadre scientifique dans lequel évolue l'application est celui de l'interaction gestuelle pour l'apprentissage du rythme. Par un simple mouvement de pince entre l'index et le pouce, de même que par l'usage de positions clés de la main, le système permet à un musicien aguerri ou débutant de créer sa propre séquence rythmique, de manière intuitive sans faire usage d'un quelconque instrument (contexte VMI [1]). Plus précisément, les gestes musicaux exploités dans notre système s'apparentent à ceux utilisés dans un dialogue de langue des signes [2], où, de la même manière, le gant de données sert à reconnaître le mouvement naturel de la main. Le gant de données étant une modalité en entrée riche en termes de nombre de patterns gestuels différents qu'il peut produire, ce dernier est exploité comme véritable tableau de bord de la production sonore.

2 Travaux antérieurs

Divers travaux importants ont déjà mis en œuvre le geste pour effectuer sa reconnaissance et/ou sa synthèse pour éventuellement le traduire en synthèse sonore. Certains d'entre eux ont exploité le gant de données ou d'autres capteurs de mouvement (Vicon®) : pour entraîner un système de reconnaissance de positions de la main associées à des sons (GRASSP [3]), pour construire une base de données de gestes élémentaires utilisée ensuite pour la synthèse de sons de percussion [4]. Dans le domaine de l'enseignement de la musique, un prototype réalisé à l'IRCAM permet, par reconnaissance de la gestuelle, l'apprentissage de la direction d'un orchestre [5]. La même idée est exploitée par [6] pour reconnaître le rythme dans la gestuelle d'un chef d'orchestre : dans les deux cas, il s'agit de capturer le mouvement de la main à l'aide d'une part d'un accéléromètre [5] et d'autre part d'une caméra [6]. Des études récentes comme Phalanger [7] exploitent la reconnaissance de gestes (réseau de neurones, SVM) pour contrôler une production musicale. Bien souvent, afin d'éviter des contraintes interactives imposées par un clavier ou une souris, le système est basé sur la reconnaissance d'images. En Interaction Homme-Machine (IHM), l'idée de remplacer la souris et le clavier par d'autres dispositifs originaux (systèmes de pointage 2D ou 3D tel que AirMouse [8] par exemple) n'est pas récent : le système Charade permettait déjà le contrôle d'une présentation sur écran par gant de données en 1993.

3 L'application musicale

Le système développé se distingue des travaux antérieurs par plusieurs points :

1. Il vise non seulement l'apprentissage de la rythmique pour un musicien (débutant ou initié) mais également le pilotage d'une application musicale complète de production, enregistrement, modification de sons de percussion, par l'intermédiaire, exclusivement, d'un gant de données sans utilisation d'aucun accessoire (contexte VMI [1]).
2. Les gestes produits par le gant de données et reconnus par le système sont de deux types :
   - séquenceurs : ces gestes déclenchent un son de percussion en lui associant un rythme (beat/sec), un type de son (caisse claire, cymbale, etc.) et une dynamique (volume sonore). Tous ces paramètres peuvent varier dans le temps,
   - contrôleurs : contrairement à GRASSP [3] où il s'agit de contrôler la voix, ces gestes permettent au musicien de contrôler la création des patterns rythmiques à savoir, le changement de l'onde sonore (effets sonores de type « attaque », « chorus », « echo », …) et le pilotage de l'enregistrement (avancer, reculer, re-jouer, enregistrer). Aucune interface d'entrée classique (clavier, souris) n'est utilisée.

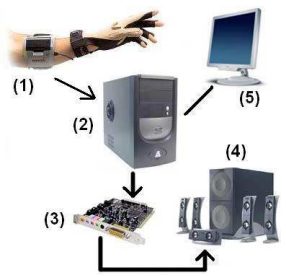


Figure 1. Schéma général du système interactif

Le gant de données (CyberGloveII®, (1) Figure 1) que nous exploitons fournit 22 mesures d'angle à une cadence de maximum 90Hz. Le schéma général de l'application musicale est celui de la Figure 1.

Pour créer ses schémas rythmiques (plus ou moins élaborés selon son niveau), l'élève en apprentissage dispose d'un gant de données (1) comme modalité d'interaction en entrée. L'application (2) reconnaît en temps réel la gestuelle capturée par le gant et restitue dans des enceintes acoustiques (4), en temps réel, les sons de percussion créés (3). Les ondes sonores sont produites grâce à la bibliothèque logicielle OpenAL (Open Audio Library) appropriée au développement d'applications sonores « temps réel ». Simultanément, le schéma rythmique (incluant un tempo de base) s'affiche à l'écran (5) sous forme d'une partition par exemple. L'élève ou le professeur peut alors l'éditer pour la modifier, la re-jouer, revenir en arrière ou avancer. Toutes ces actions sont reconnues par le système à partir de gestes spécifiques réalisés avec le gant.

4 Reconnaissance du rythme

L'utilisateur module la rythmique (c'est-à-dire le tempo de base de la partition de percussion) de façon continue : celle-ci peut donc varier au cours du temps. La méthode de reconnaissance doit répondre en temps réel pour suivre ces variations et produire un signal sonore périodique qui sera l'image du tempo reconnu. Le geste doit rester très simple et facilement modulable pour permettre une variation rapide du tempo si nécessaire. Pour répondre à ces contraintes, nous choisissons d'associer la rythmique au mouvement de battement entre un des doigts (l'index par exemple) et le pouce. L'utilisateur doit frapper régulièrement l'index contre le pouce. A chaque collision, un « beat » sonore en temps réel est produit. La fréquence avec laquelle les collisions se produisent donne le rythme au son, image du tempo reconnu.

Après observation des signaux, et dans un souci de simplification, seul l'angle de l'articulation métacarpo phalangienne (*angMCP*) du doigt est déterminant. En effet, à lui seul, il nous fournit des maximums locaux, correspondants aux collisions index/pouce, qui sont parfaitement détectables par les techniques de traitement du signal. La détection du maximum local, dans le cadre d'une application musicale, n'est acceptable que si elle se fait « au bon moment », ce qui nécessite : d'une part une reconnaissance temps réel avec décision à chaque nouvel échantillon reçu, d'autre part, une anticipation sur le signal analysé qui demande de faire une hypothèse sur la forme du mouvement futur. En effet, un délai temporel est inhérent à la production du son et un temps de latence de production de ce son supérieur à 20ms n'est généralement pas acceptable car il produit une anomalie de feedback dommageable à l'exploitation temps réel de l'interface : l'utilisateur a tendance à s'adapter à ce délais qui reste désagréable car peu naturel si le son se produit trop longtemps après l'interaction gestuelle.

Un exemple de signal généré « tel quel » par le gant de données pour *angMCP* de l'index est montré à la Figure 2. Chaque maximum local (neuf visibles à la Figure 2), correspond à un angle en degrés qui est maximum positif (position vers le bas de l'index).

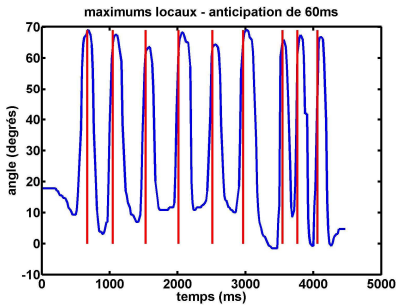


Figure 2. Détection des maximums locaux, P = 3

Les droites verticales montrent les instants de détection des 9 maximums locaux (collisions pouce/index) avec une anticipation de P = 3 échantillons (soit 60ms, 1/pe = 50Hz). Les différents tempos calculés à partir de ces 9 maximums locaux s'étendent de 1.73Hz (entre le 6ème et le 7ème maximum) à 4.59Hz (entre le 7ème et le 8ème maximum).

5 Reconnaissance du geste

Hormis la reconnaissance du rythme qui se base sur des techniques de traitement du signal (voir paragraphe précédent), les autres gestes clés de la main (exemples Figure 3) sont identifiés par une méthode de reconnaissance des formes de type logique floue [9].

STOP PLAY

Figure 3. Deux exemples de positions clés de la main

Chaque position statique de la main est représentée par une transition entre positions limites : dès lors, il est plus judicieux de construire des positions de référence plutôt que des gestes prototypes. Une (position de) référence Pat_j ($1 \leq j \leq m$, m étant le nombre de classes) est créée pour chaque geste statique à reconnaître. L'interval des positions acceptables (Pat_j) est représenté par une collection d'ensembles floues, un ensemble floue par capteur.

Etant donné une position quelconque de la main, une fonction floue d'appartenance à l'interval est évaluée pour chaque capteur. Soit C_i la valeur du capteur i ($1 \leq i \leq n$, n étant le nombre de capteurs).

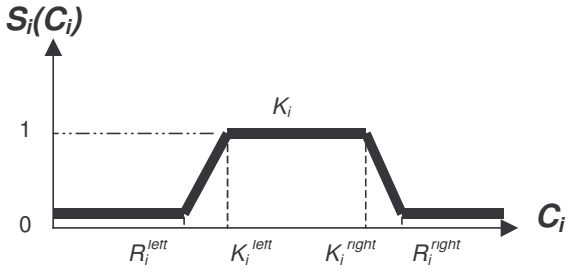


Figure 4. Fonction floue pour le capteur *i*

Comme le montre la Figure 4, la fonction floue $S_i(C_i)$ est définie par un noyau K_i et son support : les valeurs limites $[R_i^{left}\ R_i^{right}]$. Le calcul de $S_i(C_i)$ sur l'interval [0, 1] est donné par :

- $S_i(C_i) = 0$ si $C_i \in [-\infty, R_i^{left}]$ ou $C_i \in [R_i^{right}, \infty]$,
- $S_i(C_i) = 1$ si $C_i \in [K_i^{left}, K_i^{right}]$.

Une position de référence Pat_j est définie par l'ensemble des fonctions floues, chaque fonction étant associée à chacun des capteurs i : $Pat_j = \{S_{ij},\ 1 \leq i \leq n,\ n \text{ capteurs}\}$.

Apprentissage : étant donné une position de référence Pat_j ("STOP" ou "PLAY" Figure 3), le but est de construire les valeurs limites $[R_i^{left}\ R_i^{right}]$ pour chaque capteur i. Pour cela, l'utilisateur doit conserver la position statique de référence un certain temps pour permettre l'enregistrement des différentes valeurs fournies par les capteurs. En procédant de cette façon, le noyau s'étend naturellement par écartement de K_i^{left} et K_i^{right}. L'utilisateur peut recommencer l'opération autant de fois que nécessaire pour mieux affiner la forme de S_i. Les valeurs limites $[R_i^{left}\ R_i^{right}]$ quant à elles sont fixées en prenant l'inverse de l'écart type calculé sur les valeurs observées de chaque capteur i.

Reconnaissance : maintenant que les ensembles floues sont construits pour chaque position clé de la main Pat_j, le système de reconnaissance est exploitable. Une position quelconque $P(t)$ de la main est définie par l'ensemble des valeurs fournies par chaque capteur i du gant de données à l'instant t : $P(t) = \{C_i(t),\ 1 <= i <= n\}$. Tant que l'utilisateur interagit avec l'application (musicale), le logiciel calcule en continu (à chaque instant d'échantillonnage t) un score entre la position $P(t)$ et chaque référence Pat_j ($1 <= j <= m$). Pour chaque classe de reconnaissance j, le score $match\ (P(t), Pat_j)$ est donné par l'expression suivante :

$$match(P(t), Pat_j) = \frac{\sum_{i=1}^{n} \lambda_{ij} . S_{ij}(C_i(t))}{\sum_{i=1}^{n} \lambda_{ij}} \qquad (1)$$

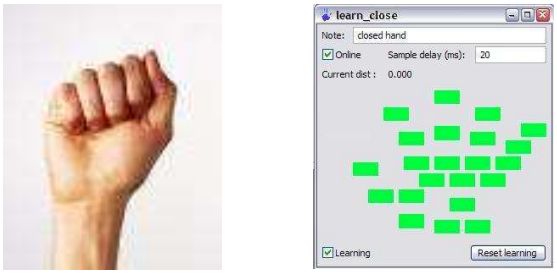
où S_{ij} est la valeur de la fonction floue du capteur i pour la référence Pat_j à l'instant t, et λ_{ij} est un coefficient réel qui joue le rôle de séparateur entre les classes. Le dénominateur de (1) permet de normaliser l'expression entre zéro et un.

La référence Pat_j ayant le meilleur score (1) avec la position de la main $P(t)$ donne la classe j reconnue, si ce score excède un seuil limite.

6 Apprentissage humain de la rythmique

Voici un exemple d'utilisation de l'application pour l'apprentissage d'un schéma rythmique par un élève. Un séquenceur tourne en boucle sur le tempo de base choisi par le professeur (voir §4) : c'est le schéma rythmique initial. Le professeur crée des agents qui serviront à reconnaître les positions clés (1 agent par position clé). Il choisit par exemple de créer un agent « learn_close » (Figure 5B) dédié à la reconnaissance d'une position clé de la main montrée à la Figure 5A (main fermée).

L'interface d'apprentissage de la position clé « main fermée » de la main est montrée à la Figure 5B.



Figures 5A et 5B. La position clé "main fermée" et l'apprentissage de cette position

Chaque rectangle (vert à la Figure 5B) représente un capteur du gant de données (22 au total). Quand le rectangle est vert, cela signifie que la valeur du capteur est compatible avec la forme apprise. Une fois l'apprentissage terminé, les rectangles sont tous verts et il est alors possible d'associer au geste (position clé) un événement qui sera un son de percussion.

Maintenant que l'ensemble des positions clés sont apprises par le système, l'élève va pouvoir les utiliser pour reproduire le plus fidèlement possible une séquence rythmique de référence préenregistrée par le professeur. Cette séquence rythmique se compose de différents éléments de percussion (caisse claire, cymbale, etc.). A l'élève de positionner correctement le bon élément au bon moment : en comparant la séquence de l'élève avec la référence, le système signale en temps réel les erreurs (cf. le jeu sonore et visuel *Simon* des années 1980).

7 Perspectives

Certains aspects intéressants de l'application musicale doivent encore être étudiés. Il s'agit d'une part de la reconnaissance de mouvements dynamiques et, d'autre part, d'aspects plus spécifiques à l'interaction homme-machine dans un contexte d'apprentissage de la musique.

Outre la synthèse de schémas rythmiques, le rôle de l'application est également de piloter les enregistrements sonores. En utilisant le gant de données comme modalité de contrôle, on offre à l'utilisateur une interaction intuitive simple. En effet, sans discontinuité, l'utilisateur pourra basculer de l'enregistrement du rythme à son contrôle. La Figure 6 montre un exemple d'un tel mouvement de contrôle : la gestuelle "bobiner/rembobiner", avec l'index, pour reculer et avancer dans l'enregistrement.

Figure 6. Gestuelle dynamique "bobiner/rembobiner"

Actuellement, la reconnaissance étant purement statique (positions clés), les mouvements dynamiques, comme celui montré à la Figure 6, ne sont pas encore traités par notre système car le geste se compose d'une succession de positions statiques que le reconnaisseur n'a pas apprises. Une évaluation de l'application musicale par des musiciens en apprentissage doit encore être organisée. Le taux de reconnaissance semble acceptable pour interagir efficacement avec l'application, mais aucune évaluation quantitative précise n'est encore disponible. L'usage d'un second gant de données doit également être étudié. Son rôle serait d'étendre l'expression gestuelle de l'utilisateur grâce à la multiplication des agents de reconnaissance.

Bibliographie

[1] Mulder, A. Virtual Musical Instruments: Accessing the sound synthesis universe as a performer. In *Proceedings of the first Brazilian Symposium on Computer Music*, Caxambu, Minas Gerais, Brazil, August 2-4, 1994.

[2] Duarte, K., Gibet, S. Heterogeneous Data Sources for Signed Language Analysis and Synthesis. In *Proceedings of the international conference on Language Resources and Evaluation (LREC 2010)*, Malta, May 2010.

[3] Pritchard, B., Fels, S. GRASSP: Gesturally-Realized Audio, Speech and Song Performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME06)*, Paris, France, June 2006.

[4] Bouënard, A., Wanderley, M.M., Gibet, S. Gesture Control of Sound Synthesis: Analysis and Classification of Percussion Gestures. *Acta Acustica united with Acustica, The Journal of the European Acoustics Association (EEA), Volume 96, Number 4*, July/August 2010.

[5] Bevilacqua, F., Guédy, F., Schnell N., Fléty E., Leroy N. Wireless sensor interface and gesture-follower for music pedagogy. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME07)*, New York, USA, June 2007.

[6] Kolesnik, P., Wanderley, M.M. Recognition, Analysis and Performance with Expressive Conducting Gestures. In *Proceedings of the International Computer Music Conference (ICMC 2004)*, Miami, USA, November, 2004.

[7] Kiefer, C., Collins, N., Fitzpatrick, G. Phalanger: Controlling Music Software With Hand Movement Using A Computer Vision and Machine Learning Approach. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME09)*, Pittsburg, USA, June 2009.

[8] Ortega, M., Nigay, L. AirMouse: Finger Gesture for 2D and 3D Interaction. In *Lecture Notes in Computer Science, LNCS 5726, Human-Computer Interaction (Interact2009), pp. 214-227*, Sweden, August 2009.

[9] Vogler, C., Tocatlidou, A. Extending Fuzzy Sets with New Evidence for Improving a Sign Language Recognition System. In *Lecture Notes in Computer Science, LNAI 5571, Fuzzy Logic and Applications*, 2009.